

Deciding Stability under FIFO in the Adversarial Queuing model in polynomial time

Maria J. Blesa

Report LSI-05-03-R

2nd February 2005



Departament de Llenguatges i Sistemes Informàtics
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Deciding Stability under FIFO in the Adversarial Queuing model in polynomial time^{*}

Maria J. Blesa

ALBCOM research group
Dept. Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Ω-213 Campus Nord, E-08034 Barcelona, Spain
mjblesa@lsi.upc.edu

Abstract. In spite of the importance of the FIFO protocol and the research efforts invested in obtaining results for it, deciding whether a given network is stable under FIFO was still an open question. In this work, we address the general case of this problem and try to characterize the property of stability under FIFO in terms of network topologies. We show that this property is decidable in polynomial time.

^{*} Partially supported by the FET Programme of the EU under contract number IST-2001-33116 (FLAGS), and by the Spanish CICYT projects TIC-2001-4917-E and TIC-2002-04498-C05-03 (TRACER). ALBCOM states for “*Algorithms, Bioinformatics, Complexity and Formal Methods*”, <http://albcom.lsi.upc.edu>.

1 Introduction

Since the emergence of computer networks, protocols were used for the establishment of ordered communications among computers. Communication takes place at different levels: low level protocols define for example the bit- and byte-ordering, their transmission, and the error detection and correction of the bit stream; high level protocols deal for example with the data packet formatting, the packet routing and the packet scheduling.

In this paper, we are interested in this latter functionality, in which the *protocol* (also called *scheduling policy*) determines the order in which the packets requiring to cross a link are scheduled to be forwarded. Most scheduling protocols aim at moving information across a network in an efficient and reliable manner. This often requires congestion and flow control, error detection and correction, and handshaking to coordinate the information transfer. Most network communication protocols are implemented as part of the operating system on the computers wishing to communicate. The *first-in-first-out* (FIFO) protocol is still one of the most popular, important and effective scheduling policies, in spite of its simplicity. The FIFO protocol schedules queued packets according to a *local criterion* in which the highest priority is given to the packet that has arrived first in the queue. This locality property makes the FIFO protocol easy to be implemented.

Appropriate models to study networking systems that implement specific communication protocols are needed. Those models could help us to understand better the dynamics of nowadays' communication networks, and therefore to detect and overcome the conditions leading to undesirable negative effects, as well as helping on their further prevention. One of those undesirable negative effects is the lack of stability.

Stability refers to the fact that the number of packets in the system remains bounded¹ as the system dynamically evolves in time. Stability is studied in relation to the three main components forming a synchronous communication system $(\mathcal{G}, \mathcal{A}, \mathcal{P})$: the network \mathcal{G} , the traffic pattern defined by \mathcal{A} , and the protocol \mathcal{P} . *Networks* are modeled by directed graphs in which the nodes represent the hosts, and the arcs represent the links between those hosts. The *traffic pattern* controls where and how packets join the system and defines their trajectory. The protocols considered are usually greedy.²

A strongest notion of stability is that of *universal stability*. Universal stability can be addressed from the network or from the protocol point of view. A network \mathcal{G} is universally stable if, for any protocol and any traffic pattern, the resulting system is stable. A protocol \mathcal{P} is universally stable if, for any network and any adversary the resulting system is stable.

According to the classification introduced in [4], we will also differentiate and refer to the property of stability in the case in which packets follow *simple paths* as *simple-path stability*, leaving then the term *stability* to refer to the case in which packets follow *paths*.³

¹ This bound, which can be a function of the system parameters, is not dependent on time.

² (Store and forward) greedy protocols are those forwarding a packet across a link e whenever there is at least one packet waiting to traverse e . Three types of packets may wait to traverse a link in a particular instant of time: the incoming packets arriving from adjacent links, the packets injected directly into the link, and the packets that could not be forwarded in previous steps. At each time step, only one packet from those waiting is forwarded through the link; the rest are kept in a queue at the link. Greedy protocols are also called work-conserving protocols.

³ We consider a *path* through a digraph a traversal of consecutive vertices along a sequence of arcs, in which repeated edges (but no vertices) within the path are permissible. When there are no repeated vertices in the path (and therefore no edges either), then it is called a *simple path*.

The Adversarial Queueing Theory (AQT) model proposed by Borodin et al. [7] has become an important model to study stability issues in packet-switched communication networks. These models have been shown to be good theoretical frameworks for describing the traffic pattern in both connectionless networks (such as the Internet) and short-term connection networks, as well as connection-oriented networks (such as ATM networks). Adversarial models allow to analyze the system in a worst-case scenario, since they have replaced traditional stochastic arrival assumptions in the traffic pattern by worst-case inputs. The AQT model considers the time evolution of a packet-routing network as a game between an adversary, which produces the traffic pattern, and a queueing policy. The system is considered to be *synchronous*. At each time step the adversary may inject a set of packets to some of the nodes. For each packet, the adversary specifies the route that it must traverse (static routing) before arriving to its destination and disappear from the system. If more than one packet wishes to cross an edge e at the same time step, then the queueing policy chooses exactly one of these packets. The remaining packets wait in the queue. This game then advances to the next time step. The goal of the adversary is to try to prevent the protocol from guaranteeing load and delay bounds. On the contrary, the main goal of the model is to study conditions for stability of the network under different protocols.

In order not to trivially overload the system and in order to be able to guarantee delay bounds, it is necessary to restrict the traffic arriving to the network. The constraints on the traffic pattern must ensure that, over long periods of time, the maximum traffic injected in a link is roughly the amount of traffic that the link can forward. Two parameters (r, b) constraint an adversary in the AQT model, where $b \geq 0$ is the *burstiness* and $0 < r < 1$ is the *injection rate*. Let $N_e(I)$ be the number of packets injected by the adversary in a time interval I , whose path require to traverse a particular edge e . The adversary must obey the following (leaky-bucket) constraint:

$$N_e(I) \leq \lceil r|I| \rceil + b. \quad (1)$$

Recent research on stability has mainly considered the AQT model and has put special interest in the FIFO protocol (see, e.g., [7,5,8,17,18,9,6,19]).

Our motivations and contributions. Universal stability of networks is a non-trivial property; since it is a predicate quantified over all protocols and adversaries, it might at first appear that it is not a decidable property. One of the deepest results in the context of network stability in the adversarial queueing model establishes that, to the contrary, this is not the case [5]. The question of characterizing networks that are universally stable, and algorithmically recognizing such networks, naturally arises next. This question was also recently answered in [4] by fully characterizing the property under different network representation and considering different restrictions on the packet trajectories. Moreover, in the same work it is shown that deciding universal stability of networks requires polynomial time.

Concerning the protocol point of view, it is known that FTG, NFS, SIS and LIS are universally stable, while FIFO, LIFO, NTG and FFS are not [5].⁴ For those queueing policies which are not universally stable, a weaker notion of stability is addressed, that of the *stability under a protocol*.

⁴ The protocol *last-in-first-out* (LIFO) gives priority to the packet which entered the queue the latest. Concerning injection times, *shortest-in-system* (SIS) gives priority to the packet introduced last into the system, while *longest-in-system* (LIS) gives it to the one that has been in the system the longest. Concerning the distance to the destination, *nearest-to-go* (NTG) assigns highest priority to the packet that is closest to its destination and FTG (Furthest To Go) to the packet that is farthest. Similarly, *nearest-from-source* (NFS) and *farthest-from-source* (FFS) consider the distance to the source.

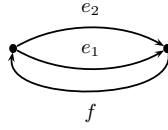


Fig. 1. Digraph \mathcal{U}_1 for which still many questions about its stability are not solved. The (open) characterization proposed in this work revolves around it.

Here the problem is to decide which networks are stable, and which are not, under a fixed queuing policy \mathcal{P} . In the best case, a characterization of stability under the protocol \mathcal{P} can be obtained. To the best of our knowledge, only two results are known in this sense: Deciding stability under the NTG-LIS⁵ and FFS protocols is polynomially solvable and it is, moreover, equivalent to deciding universal stability of networks [4,1].

In this paper *we address the problem of deciding stability under the FIFO protocol*. In spite of the importance of this property and this protocol, the aspects concerning its decidability and complexity were still an open question. In this work, *we show that the property of stability under FIFO is decidable in polynomial time*.

Taking the characterization of (network) universal stability as starting reference, we propose an (open) characterization of the stability under FIFO. The characterization is composed by two candidate sets of forbidden subgraphs. The eligibility of one or the other candidate set as the decisive characterization depends on the stability of the digraph \mathcal{U}_1 (see Figure 1). In the case that \mathcal{U}_1 is unstable under FIFO, the characterization would be the same as the characterization of the digraphs that are universally stable [4]. This would have some nice implications since, in the case this holds, a digraph would be universally stable if and only if it is stable under FIFO. In spite of the simplicity of the network topology in \mathcal{U}_1 , some important questions about it remain still open nowadays. One of these particular questions is concerned with its stability under the FIFO protocol.

Organization. The paper is organized as follows. In Section 2 we introduce some preliminaries of the work; this includes a review of the results existing in the literature which are concerned with stability under FIFO, and also the notation used in the forthcoming of the paper. In Section 3, the family of digraphs which are stable under FIFO are presented. Also in this section, the family of digraphs which are not stable are introduced by its minimal representants. The property of stability under FIFO is characterized in Section 4 in terms of those unstable minimal representants. In the same section, a polynomial-time algorithm is given for deciding the property of stability under the FIFO protocol. The work concludes in Section 5, where some open questions as well as some possible extensions of the work are pointed out.

Most proofs of the lemmas and theorems in this work are posted as appendix.

2 Preliminaries

The *first-in-first-out* (FIFO) greedy protocol is probably one of the most commonly used scheduling protocols. FIFO is used in many contexts in computer environments, either internally (e.g., in operating systems to process I/O device interruptions or information exchange between processes) or externally (e.g., as communication protocol for information exchange between computers). One of its main advantages, specially when implementing it, is that its criterion to

⁵ The protocol NTG-LIS works as NTG, but solves ties using the LIS protocol.

schedule packets is completely based on local properties. In the FIFO protocol, highest priority is given to the packet that has arrived first in the queue. Observe that, when two packets arrive to the queue at the same time then they have to be queued in some order, which we will assume that is decided *arbitrarily* by the adversary.

2.1 Previous results on FIFO in the AQT model

Due to its relevance, much attention has been put on the study of stability conditions in AQT under the FIFO protocol. Already the pioneering work of Borodin et al. [7] showed that ring topologies are not stable under FIFO for the extreme injection rate $r = 1$. It is however of higher interest to find bounds when adversaries work in underloaded conditions, i.e., when their injection rate $r < 1$. Thus, the consecutive improvement of the lower bounds for instability under FIFO was one of the research subjects in the last years. As time and research advanced, this lower bound was dropping from $r \geq 0.85$ [5], to $r \geq 0.84$ [11], $r \geq 0.8357$ [8], $r \geq 0.771$ [17], $r \geq 0.749$ [16], and finally to $r > 0.5$ [19]. A further step was done recently, when FIFO was shown to be unstable at arbitrarily low rates [6,15]. Meanwhile, the upper bound for the stability of FIFO was also improved from $r < 1/9$ [17] to $r < 0.1428$ [8]. The existence of a network-dependent upper bound was shown in [8], which was recently generalized to $r \leq 1/d$ [19,9], where d is the length of the longest route traversed by any packet.

In spite of the importance of this protocol and all these results, deciding whether a given network is stable under FIFO remains still an open question in many cases. This work aims at doing a step forward into solving the problem of deciding stability in the AQT model under the FIFO protocol and study its complexity. As we have said before, the decisive characterization depends on the stability of the digraph \mathcal{U}_1 (see Figure 1).

2.2 Graph subdivision operations

In the following, we use standard graph terminology to denote the following digraphs: directed k -cycles, *acyclic* digraphs, and *unicyclic* digraphs. A directed k -cycle is a directed cycle with k vertices, where $k \geq 2$. A unicyclic digraph is a digraph that contains only one cycle.

We will characterize the property of stability under FIFO in terms of a family of forbidden subgraphs. To this aim, we first need to identify the families of digraphs which are stable under this protocol. Then the simplest digraphs which are not stable should be identified. The family of the digraphs which are not stable under FIFO will be then defined by iteratively applying *subdivision operations* to those simplest digraphs. We consider the following subdivision operations:

- The *subdivision of an arc* (u, v) in a digraph \mathcal{G} consists in the addition of a new vertex w and the replacement of (u, v) by the two arcs (u, w) and (w, v) .
- The *subdivision of a 2-cycle* $(u, v), (v, u)$ in a digraph \mathcal{G} consists in the addition of a new vertex w and the replacement of $(u, v), (v, u)$ by the arcs $(u, w), (w, u), (v, w)$ and (w, v) .

Then, given a digraph \mathcal{G} , we will denote as $\mathcal{E}(\mathcal{G})$ the family of digraphs formed by \mathcal{G} and all the digraphs obtained from \mathcal{G} by successive arc or 2-cycle subdivisions. Note that, a strongly connected digraph remains so when applying arc or 2-cycle subdivisions to it.

In the following, we will be using digraphs and networks as synonyms. All the digraphs considered in this paper are strongly connected and they may have multiple edges (arcs) but no

loops.⁶ We consider that a packet transmitted over those digraphs follows a predefined *path*. To keep lighter the notation, a path is specified by the sequence of its edges or by the concatenation of subpaths. Moreover, the names used to denote the digraphs and their edges correspond to the ones depicted in Figure 2.

3 Stability of digraphs under FIFO

In this section, we show which digraphs are stable under FIFO as well as those simplest digraphs which are not stable under this protocol. By applying subdivision operations to those simplest unstable digraphs the whole family of digraphs which are not stable under FIFO will be determined.

All directed acyclic graphs⁷ and (isolated) directed cycles on any number of vertices are known to be universally stable [7,5], thus being also stable under the FIFO protocol. Let us re-write this consequence as Lemma 1.

Lemma 1 ([7,5]). *All acyclic digraphs and k -cycles (where $k \geq 2$) are stable under FIFO.*

This property is maintained when acyclically connecting digraphs which are stable under FIFO. Given two digraphs \mathcal{G}_1 and \mathcal{G}_2 , let us denote as $\mathcal{G}_{1 \rightarrow 2}$ the family of digraphs formed by joining \mathcal{G}_1 and \mathcal{G}_2 with arcs that go only from \mathcal{G}_1 to \mathcal{G}_2 .

Lemma 2. *If digraphs \mathcal{G}_1 and \mathcal{G}_2 are stable under FIFO, then so is any graph $\mathcal{G} \in \mathcal{G}_{1 \rightarrow 2}$.*

Proof. Assume that the adversary working against \mathcal{G} has rate r and burstiness b . Any packet injected into \mathcal{G}_1 by this adversary will get out of \mathcal{G}_1 within a bounded number of time steps t_1 , because \mathcal{G}_1 is stable under FIFO. Some of the packets leaving \mathcal{G}_1 might join \mathcal{G}_2 . Let us consider a time interval of t_2 steps starting right after the t_1 time steps mentioned before. The packets joining \mathcal{G}_2 during that t_2 time steps must have been introduced in the system during the last $t_1 + t_2$ steps; moreover, there are at most $r(t_1 + t_2) + b$ of those packets.

We want to show that all the packets coming from \mathcal{G}_1 together with the packets injected directly in \mathcal{G}_2 could have been generated by an adversary working only against \mathcal{G}_2 . Consider that such an adversary has rate $1 > r' > r$ and burstiness $b' \geq b$. During any interval $t \geq t_2$, the total amount of packets introduced into \mathcal{G}_2 would be $r't + b'$. In order for those packets to be generated by the mentioned adversary, it must hold that $r't + b' = r't + b' - ((r' - r)t_2 + b' - rt_1 - b)$, which holds when considering $t_2 = rt_1 / (r' - r)$. ■

As a consequence of the previous lemma we have,

Corollary 1. *All unicyclic digraphs are stable under FIFO.*

Corollary 2. *A digraph \mathcal{G} with more than one cycle is stable under FIFO if and only if no pair of cycles shares vertices, i.e., if and only if the cycles are acyclically connected by directed paths.*

Theorem 1. *A digraph \mathcal{G} is stable under FIFO if and only if all its strongly connected components are stable under FIFO.*

⁶ Multiple edges share the same pair of different endpoints. The endpoints of a loop is the same vertex.

⁷ Note that this includes directed *trees* and *multi-trees*, i.e., directed trees with single arcs and multi-arcs.

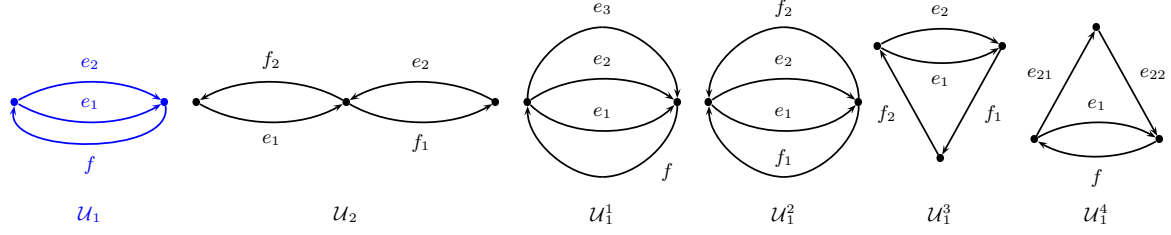


Fig. 2. Minimum forbidden subgraphs characterizing stability under FIFO. The two candidate sets to consider are either $\{\mathcal{U}_1, \mathcal{U}_2\}$ or $\{\mathcal{U}_1^1, \mathcal{U}_1^2, \mathcal{U}_1^3, \mathcal{U}_1^4, \mathcal{U}_2\}$; the former would characterize the stability under FIFO in the case that \mathcal{U}_1 is not stable under that protocol, while the latter would characterize it in the case that \mathcal{U}_1 is stable under FIFO.

In a strongly connected digraph, every vertex can be accessed from any other vertex of the digraph. Note that all the directed acyclic digraphs as well as all the digraphs formed by acyclic connections are not strongly connected. However, in the context of communication networks strongly connected topologies are of highest interest. Beyond the directed cycle, the next networks to consider are then the digraphs \mathcal{U}_1 and \mathcal{U}_2 depicted in Figure 2, which are the smallest non-unicyclic strongly connected digraphs, i.e., the smallest strongly-connected digraphs with more than one cycle. In the following, we show that neither the digraph \mathcal{U}_2 , nor any of its extensions, are stable under FIFO.

Lemma 3. *The digraphs in $\mathcal{E}(\mathcal{U}_2)$ are not stable under FIFO.*

However, it remains still an open question (as it was already pointed out in [4]) whether the digraph \mathcal{U}_1 is stable under FIFO. Instead, let us consider the digraphs \mathcal{U}_1^1 , \mathcal{U}_1^2 , \mathcal{U}_1^3 and \mathcal{U}_1^4 depicted in Figure 2, which are the next strongly connected digraphs to consider after \mathcal{U}_1 (in terms of their size). Digraphs \mathcal{U}_1^1 and \mathcal{U}_1^2 are obtained from \mathcal{U}_1 when considering multi-edges, while digraphs \mathcal{U}_1^3 and \mathcal{U}_1^4 are obtained from \mathcal{U}_1 when subdividing arcs. The digraphs that result from 2-cycle subdivisions of \mathcal{U}_1 contain \mathcal{U}_2 as a subgraph, and so they can be made unstable under the FIFO protocol. Although no result concerning the stability of \mathcal{U}_1 under FIFO is known, we show in the following that neither digraphs \mathcal{U}_1^1 , \mathcal{U}_1^2 , \mathcal{U}_1^3 or \mathcal{U}_1^4 , nor any of their extensions are stable under FIFO.

Lemma 4. *The digraphs in $\mathcal{E}(\mathcal{U}_1^1) \cup \mathcal{E}(\mathcal{U}_1^2) \cup \mathcal{E}(\mathcal{U}_1^3) \cup \mathcal{E}(\mathcal{U}_1^4)$ are not stable under FIFO.*

However, a quite high injection rate ($r \geq 0.929$, see appendix) is needed to produce instability in this networks, which indicates that, although possible, it is not “easy” to make a system unstable under FIFO with these underlying topologies. This behavior was already observed while trying to improve the lower bound for the instability of the FIFO protocol [5,11,8,17,16,19].

Observe that, by considering the family of digraphs composed by $\mathcal{U}_2, \mathcal{U}_1^1, \mathcal{U}_1^2, \mathcal{U}_1^3, \mathcal{U}_1^4$ and their extensions, the only graphs which are not included are those which have as subgraph a graph in $\mathcal{E}(\mathcal{U}_1) \setminus \{\mathcal{E}(\mathcal{U}_2) \cup \mathcal{E}(\mathcal{U}_1^1) \cup \mathcal{E}(\mathcal{U}_1^2) \cup \mathcal{E}(\mathcal{U}_1^3) \cup \mathcal{E}(\mathcal{U}_1^4)\}$, i.e., those whose strongly connected components are exactly \mathcal{U}_1 . If \mathcal{U}_1 is stable under FIFO, then those digraphs are also because of Lemma 2; if \mathcal{U}_1 is not stable under FIFO, then those digraphs can be made also unstable but, in this case, they would not be the smallest forbidden subgraphs because they contain \mathcal{U}_1 .

4 Characterizing stability under FIFO

In this section two candidate sets of forbidden subgraphs are proposed for the characterization of the stability under FIFO. The choice of the right candidate has a penchant for one subset or the other depending on the stability of \mathcal{U}_1 . Whatever the decisive characterization is, we can state the stability under FIFO can be decided in polynomial time.

Theorem 2. *If the digraph \mathcal{U}_1 is not stable under FIFO, then any digraph \mathcal{G} is stable under FIFO if and only if it does not contain as subgraph a digraph from $\mathcal{E}(\mathcal{U}_1) \cup \mathcal{E}(\mathcal{U}_2)$. Otherwise, if the digraph \mathcal{U}_1 is stable under FIFO, then any digraph \mathcal{G} is stable under FIFO if and only if it does not contain as subgraph a digraph from $\mathcal{E}(\mathcal{U}_1^1) \cup \mathcal{E}(\mathcal{U}_1^2) \cup \mathcal{E}(\mathcal{U}_1^3) \cup \mathcal{E}(\mathcal{U}_1^4) \cup \mathcal{E}(\mathcal{U}_2)$.*

Proof. If the digraph \mathcal{U}_1 is not stable under FIFO then, according to Theorem 3 and the fact that the instability of a subgraph implies the instability of the whole digraph, no digraph \mathcal{G} containing as subgraph a digraph from $\mathcal{E}(\mathcal{U}_1) \cup \mathcal{E}(\mathcal{U}_2)$ is not stable. If \mathcal{G} does not contain as subgraph a digraph from $\mathcal{E}(\mathcal{U}_1) \cup \mathcal{E}(\mathcal{U}_2)$ then all its strongly connected components must consist of at most one simple directed cycle. Therefore, \mathcal{G} is stable under FIFO according to Lemma 1 and Theorem 1.

If the digraph \mathcal{U}_1 is stable under FIFO then, according to Lemmas 3 and 4, together with the fact that the instability of a subgraph implies the instability of the whole digraph, no digraph \mathcal{G} containing as subgraph a digraph from $\mathcal{E}(\mathcal{U}_1^1) \cup \mathcal{E}(\mathcal{U}_1^2) \cup \mathcal{E}(\mathcal{U}_1^3) \cup \mathcal{E}(\mathcal{U}_1^4) \cup \mathcal{E}(\mathcal{U}_2)$ is stable. If, on the contrary, \mathcal{G} does not contain as subgraph a digraph from that set, then all its strongly connected components either consist of at most one simple directed cycle (and then, according to Lemma 1 and Theorem 1, \mathcal{G} would be stable under FIFO), or they contain as a subgraph the digraph \mathcal{U}_1 (which here we have assumed that is stable under FIFO). ■

This result can be stated in terms of digraphs' properties.

Corollary 3. *If the digraph \mathcal{U}_1 is not stable under FIFO, then a strongly connected digraph \mathcal{G} is stable under FIFO if and only if \mathcal{G} is a directed k -cycle (on any number of vertices $k \geq 2$). Otherwise, if the digraph \mathcal{U}_1 is stable under FIFO, then a strongly connected digraph \mathcal{G} is stable under FIFO if and only if \mathcal{G} is a directed k -cycle ($k \geq 3$) or a 2-cycle with at most one multi-edge.*

Then, instead of detecting the proposed forbidden subgraphs by means of subgraph homeomorphism (which would be NP-complete [10]), the stability of digraphs under FIFO can be decided in polynomial time by detecting the proposed forbidden subgraphs in terms of the digraphs' properties outlined in Corollaries 1, 2, and 3.

Theorem 3. *The stability under FIFO of a given digraph can be decided in polynomial time.*

Proof. Algorithms 1 and 2 check stability under FIFO of a given strongly connected digraph \mathcal{G} according to Corollaries 1, 2, and 3. The total execution time of each of the algorithm is polynomial. Algorithm 1 would be applied in the case that \mathcal{U}_1 is not stable under FIFO, while Algorithm 2 would be applied in the case that \mathcal{U}_2 is stable under FIFO.

According to Theorem 1, the strongly connected components of the digraph need to be computed first. Thus the result follows by combining the computation of the strongly connected components of the given digraph (which requires polynomial time) either with Algorithm 1, or with Algorithms 2. ■

Algorithm 1 : Stability under FIFO (sup. \mathcal{U}_1 is not stable under FIFO)

INPUT: A strongly connected digraph $\mathcal{G} = (V, E)$
 compute a directed k -cycle $C = (e_1 \dots e_k)$ of $k \geq 2$ vertices, and let
 $C_V \leftarrow \{v \mid \exists e \in C : e = (v, u)\} \subseteq V$ (set of vertices of the cycle C)
 $C_E \leftarrow \{e \mid e \in C\} \subseteq E$ (set of edges of the cycle C)

if \mathcal{G} does not have a directed k -cycle of $k \geq 2$ vertices **then**
 return YES
else
 Let $\mathcal{G}' = (V, E \setminus C_E)$ be the digraph resulting after removing from \mathcal{G} the arcs in C
 if there are two different vertices $u, v \in C_V$ connected in \mathcal{G}' by a directed path **then**
 return NO
 end if
end if

Algorithm 2 : Stability under FIFO (sup. \mathcal{U}_1 is stable under FIFO)

INPUT: A strongly connected digraph $\mathcal{G} = (V, E)$
 compute a directed k -cycle $C = (e_1 \dots e_k)$ of $k \geq 2$ vertices, and let
 $C_V \leftarrow \{v \mid \exists e \in C : e = (v, u)\} \subseteq V$ (set of vertices of the cycle C)
 $C_E \leftarrow \{e \mid e \in C\} \subseteq E$ (set of edges of the cycle C)

if \mathcal{G} does not have a directed k -cycle of $k \geq 2$ vertices **then**
 return YES
else
 Let $\mathcal{G}' = (V, E \setminus C_E)$ be the digraph resulting after removing from \mathcal{G} the arcs in C
 if there are two different vertices $u, v \in C_V$ connected in \mathcal{G}' by a directed path **then**
 $P \leftarrow$ such a directed path connecting $u \in C_V$ and $v \in C_V$ in \mathcal{G}'
 if $|C_V| = k > 2$ **then**
 return NO
 else if $|P| > 1$ **or** there is another directed path $P' \neq P$ in \mathcal{G}' between two different vertices in C_V **then**
 return NO
 end if
 end if
 compute the strongly connected components of \mathcal{G}'
 if a strongly connected component of \mathcal{G}' contains a directed k -cycle of $k \geq 2$ vertices **then**
 return NO
 else
 return YES
 end if
end if

5 Conclusions, remarks, open questions and further work

In spite of the importance of the FIFO protocol and the research efforts invested in obtaining results for it, deciding whether a given network is stable under FIFO was still an open question. In this work, we have addressed this problem and tackled the general case, i.e., the decidability and complexity of stability under the FIFO protocol. In this work, we have shown that the property of stability under FIFO is decidable in polynomial time.

We wanted to identify which network topologies determine that a system under FIFO is (or is not) stable and then, be able to provide a characterization of the property. Taking the characterization of (network) universal stability as starting reference, we have proposed an (open) characterization of the stability under FIFO (see Theorem 2). The characterization is composed

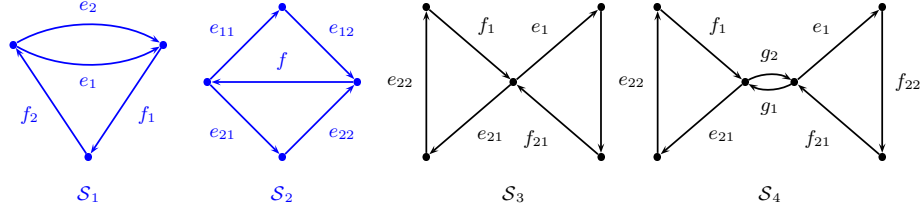


Fig. 3. Minimum forbidden subgraphs characterizing simple-path universal stability [4]. The digraphs \mathcal{S}_3 and \mathcal{S}_4 will belong to the set of forbidden subgraphs characterizing the property of simple-path stability under FIFO. In order to know which digraphs complete that characterization, the stability under FIFO of the digraphs \mathcal{S}_1 and \mathcal{S}_2 , their extensions, and the digraphs with the same basic topology but multi-edges, need to be studied.

by two candidate sets of forbidden subgraphs. The eligibility of one or the other candidate set as the decisive characterization depends on the stability of the digraph \mathcal{U}_1 (see Figure 1). In the case that \mathcal{U}_1 is unstable under FIFO, the characterization would be defined by \mathcal{U}_1 and \mathcal{U}_2 , and it would be the same as the characterization of the digraphs that are universally stable [4]. This would have some nice implications since, in the case this holds, a digraph would be universally stable if and only if it is stable under FIFO. In the case that \mathcal{U}_1 is stable under FIFO, the characterization would be defined by \mathcal{U}_2 , \mathcal{U}_1^1 , \mathcal{U}_1^2 , \mathcal{U}_1^3 , and \mathcal{U}_1^4 .

Some important questions remain still open concerning the stability under FIFO, being of course the most important one that of finding out *whether \mathcal{U}_1 can be made unstable under FIFO*, which would establish the decisive characterization.

In the same way as we proceeded in this work, other variants of stability can be tackled. Different variants can be defined according to the constraints on the packet trajectory and, as it was shown in [4], this influences strongly the characterization of the stability properties. Keeping the representation of the network as a directed graph, we can consider also the property of *simple-path* stability under FIFO. *The characterization of this property is also an open question nowadays.* A first step to it would be to study what is the behaviour of the smallest digraphs which are known not to be universally stable, when the system schedules the packets according to the FIFO policy. Those digraphs are exactly the ones depicted in Figure 3, which characterize the universal stability of networks [4]. The following lemma states the simple-path instability under FIFO of the digraphs in $\mathcal{E}(\mathcal{S}_3) \cup \mathcal{E}(\mathcal{S}_4)$.

Lemma 5. *The digraphs in $\mathcal{E}(\mathcal{S}_3) \cup \mathcal{E}(\mathcal{S}_4)$ are not simple-path stable under FIFO.*

This is a first step into the characterization of the property of simple-path stability under FIFO, however the simple-path stability of the digraphs in $\mathcal{E}(\mathcal{S}_1) \cup \mathcal{E}(\mathcal{S}_2)$, together with other digraphs which do not contain any digraph in $\mathcal{E}(\mathcal{S}_3) \cup \mathcal{E}(\mathcal{S}_4)$ as a subgraph, have to be deeply studied before converging to a characterization of the property.

The characterization of stability (and variants) under protocols other than FIFO are still also an open question in this topic. To the best of our knowledge, only the characterization of stability (and variants) under FFS and NTG-LIS are additionally known [1,4]. Probably, establishing the characterization of the stability under LIFO would be of higher interest, because the LIFO protocol is gaining popularity in the last years due to the discovery of the significant quality improvement on the performance of interactive real-time services, such as IP telephony and IP teleconferencing (and, in general, any voice and video transmission), when the network is congested [12,13,14].

References

1. C. Àlvarez, M. Blesa, J. Díaz, A. Fernández, and M. Serna. The complexity of deciding stability under FFS in the adversarial model. *Information Processing Letters*, 90(5):261–266, 2004.
2. C. Àlvarez, M. Blesa, J. Díaz, A. Fernández, and M. Serna. Adversarial models for priority-based networks. *Networks*, 45(1):23–35, 2005.
3. C. Àlvarez, M. Blesa, and M. Serna. Universal stability of undirected graphs in the adversarial queueing model. In *14th ACM Symposium on Parallel Algorithms and Architectures (SPAA'02)*, pages 183–197, Winnipeg, Canada, 2002. ACM Press New York, USA.
4. C. Àlvarez, M. Blesa, and M. Serna. A characterization of universal stability in the adversarial queueing model. *SIAM Journal on Computing*, 34(1):41–66, 2004. Partial preliminary work in [3].
5. M. Andrews, B. Awerbuch, A. Fernández, J. Kleinberg, T. Leighton, and Z. Liu. Universal stability results for greedy contention-resolution protocols. *Journal of the ACM*, 48(1):39–69, 2001.
6. R. Bhattacharjee, A. Goel, and Z. Lotker. Instability of FIFO at arbitrarily low rates in the adversarial queueing model. *SIAM Journal on Computing*, 34(2):318–332, 2004.
7. A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. Williamson. Adversarial queueing theory. *Journal of the ACM*, 48(1):13–38, 2001.
8. J. Díaz, D. Koukopoulos, S. Nikolettseas, M. Serna, P. Spirakis, and D. Thilikós. Stability and non-Stability of the FIFO Protocol. In *13th annual ACM Symposium on Parallel Algorithms and Architectures (SPAA'01)*, pages 48–52, Crete Island, Greece, 2001.
9. J. Echagüe, V. Cholvi, and A. Fernández. Universal stability results for low rate adversaries in packet switched networks. *IEEE Communication Letters*, 7(12):578–580, 2003.
10. S. Fortune, J. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, 10(4):111–121, 1980.
11. A. Goel. Stability of networks and protocols in the adversarial queueing model for packet routing. *Networks*, 37(4):219–224, 2001.
12. M. Hamdi. LCFS queueing for real-time audio-visual services in packet networks. In *IEEE International Workshop on Audio-Visual Services over Packet Networks (AVSPN'97)*, Aberdeen, Scotland, UK, 1997.
13. M. Hamdi, R. Noro, and J.-P. Hubaux. *Fresh Packet First* scheduling for voice traffic in congested networks. Technical Report SSC034, Computer Science Department, Swiss Federal Institute of Technology (EPFL), 1997. Also as US Patent Application.
14. M. Hamdi. *Fresh Packet First* scheduling for interactive services in the Internet. in *4th International Conference on Information Systems Analysis and Synthesis (ISAS'98)*. Orlando, Florida, USA, 1998.
15. D. Koukopoulos, M. Mavronicolas, and P. Spirakis. FIFO is unstable at arbitrarily low rates (even in planar networks). *Electronic Colloquium on Computational Complexity*, 10(16), 2003.
16. D. Koukopoulos, S. Nikolettseas, and P. Spirakis. The range of stability for heterogeneous and FIFO queueing networks. *Electronic Colloquium on Computational Complexity*, TR01-099, 2001.
17. D. Koukopoulos, S. Nikolettseas, and P. Spirakis. Stability behavior of FIFO protocol in the adversarial queueing model. In A. K. Y. Manolopoulos, S. Evripidou, editor, *8th Panhellenic Conference on Informatics (PCI'2001)*, volume 2563 of *Lecture Notes in Computer Science*, Nicosia, Cyprus, 2001. Springer-Verlag.
18. D. Koukopoulos, S. Nikolettseas, and P. Spirakis. Stability issues in heterogeneous and FIFO networks under the adversarial queueing model. In *8th International Conference on High Performance Computing (HIPC'01)*, volume 2228 of *Lecture Notes in Computer Science*, pages 3–14, Hyderabad, India, 2001. Springer-Verlag Heidelberg.
19. Z. Lotker, B. Patt-Shamir, and A. Rosén. New stability results for adversarial queueing. *SIAM Journal on Computing*, 33(2):286–303, 2004.

A Appendix

All the proofs of instability in this appendix are based on induction. A set of rounds compose a step of the induction reasoning. The goal is to demonstrate that the number of packets in the system can increase from step to step (and, by applying the inductive hypothesis, they can increase infinitely). The configuration of the system at the end of every step must be the same as at the beginning (in terms of the type and the location of the packets). For the sake of simplicity, we only reproduce the inductive step and sometimes we omit some additive constants in our analysis, however, those omissions will not change the final result.

A.1 Instability under FIFO

Digraphs \mathcal{U}_1^1 and \mathcal{U}_1^2 are obtained from \mathcal{U}_1 when considering multiedges, while digraphs \mathcal{U}_1^3 and \mathcal{U}_1^4 are obtained from \mathcal{U}_1 when subdividing arcs. The digraphs that result from 2-cycle subdivisions of \mathcal{U}_1 contain \mathcal{U}_2 as a subgraph, and so they can be made unstable under the FIFO protocol using the strategies used for the digraphs in $\mathcal{E}(\mathcal{U}_2)$.

For clarity reasons, we split the proof of Lemma 4 in the following four lemmas (Lemmas 7 to 10), one for each minimum forbidden subgraph derived from \mathcal{U}_1 and its corresponding family of extensions. Moreover, in each of the lemmas, we first prove that the smallest digraph \mathcal{G} of the family can be made unstable under the FIFO protocol and then, we show that the systems with a network which is a proper extension of it, i.e., $\mathcal{G}' \in \mathcal{E}(\mathcal{G}) \setminus \mathcal{G}$, can also be made unstable under the FIFO protocol. By showing first the instability in the smallest forbidden digraphs, one gets a better intuition on how is the strategy of the adversary and which contention power it has. The strategy used when the network is a proper extension of the smallest forbidden digraph is usually the natural extension to paths of the strategy used there. The only requirement that the fact of dealing with an network extension imposes is that, the initial configuration of the system has to have enough initial packets (the quantity will depend on the length of the edge extensions) to allow the accumulation.

Lemma 6. *The digraphs in $\mathcal{E}(\mathcal{U}_2)$ are not stable under FIFO.*

Proof. As a general result, the digraph \mathcal{U}_2 was already shown in [2] not to be stable in the AQT model under FIFO via an adversary with injection rate $r \geq 0.914$.

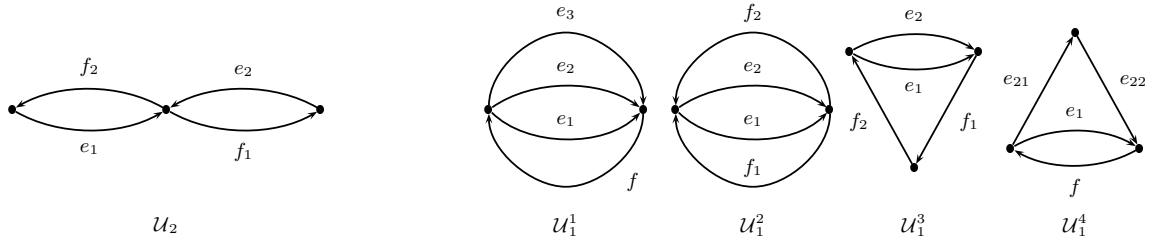
Digraphs in $\mathcal{E}(\mathcal{U}_2)$: Let \mathcal{G} be a proper extension of the graph \mathcal{U}_1^2 , as described at Figure 4(a). Let us denote with a dashed style not an arc but a directed path. Arcs are labelled with latin letters, while paths are labelled with greek letters. Moreover $|\alpha| \geq 1$ and $|\beta| \geq 1$.

At the beginning there are s packets queued in f_1 requiring to traverse only edge f_1 . Then the adversary will play infinitely the following rounds:

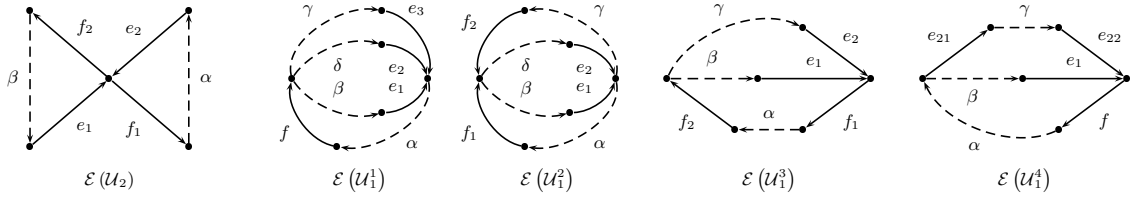
Round 1: for s steps, the adversary injects rs packets of the form $(f_1\alpha e_2 f_2\beta e_1)$. These injections will all get blocked in f_1 .

Round 2: for the next rs steps, the adversary injects r^2s packets of the form $(f_1\alpha e_2)$ and r^2s packets of the form $(f_2\beta e_1)$. The former are blocked at f_1 , while the latter get mixed with the packets flowing from the previous round, and a total amount of $r^2s - r|\alpha|$ packets are kept in the queue of f_2 .

Round 3: for the next r^2s steps, the adversary injects r^3s packets of the form (αe_2) and r^3s packets of the form $(\beta e_1 f_1)$. Both get mixed with the packets flowing from the previous round.



(a) Minimum forbidden subgraphs characterizing stability under FIFO with path trajectories. The sets to consider are either $\{\mathcal{U}_1, \mathcal{U}_2\}$ or $\{\mathcal{U}_1^1, \mathcal{U}_1^2, \mathcal{U}_1^3, \mathcal{U}_1^4, \mathcal{U}_2\}$, depending on the stability of \mathcal{U}_1 under FIFO.



(b) Extensions of the digraphs in Figure 4(a). Let us denote with a dashed style not an arc but a directed path. Arcs are labelled with latin letters, while paths are labelled with greek letters. By definition, $|\alpha| \geq 0$, $|\beta| \geq 0$, and $|\gamma| \geq 0$; in a proper extension, $|\alpha| \geq 1$, $|\beta| \geq 1$, and $|\gamma| \geq 1$.

Fig. 4. Family of forbidden subgraphs characterizing stability under FIFO.

In the first edge of α the queue has total size r^3s and, in the first queue of β the queue has total size $r^3s - r|\alpha|$.

Round 4: for the next r^3s steps, the adversary injects r^4s packets of the form (e_2f_1) and r^4s more of the form (e_1) . The former get mixed at e_2 with packets from the previous round, and a total amount of $r^4s - r|\alpha|$ mixed packets stay in the queue of e_2 . The latter get also mixed with packets from the previous round, and a total amount of $r^4s - r(|\alpha| + |\beta|)$ remain queued at e_1 .

Round 5: for the next $r^4s - r|\alpha|$ steps, the adversary injects $r^5s - r^2|\alpha|$ packets with path (f_1) .

At the end of the fifth round, there are at least $r^5s + r^4s - r^2|\alpha| - r(|\alpha| + |\beta|)$ packets queued in f_1 and requiring to traverse only that edge. The adversary described above uses path trajectories and makes any digraph $\mathcal{G} \in \mathcal{E}(\mathcal{U}_2) \setminus \mathcal{U}_2$ unstable when

$$r^5s + r^4s - r^2|\alpha| - r(|\alpha| + |\beta|) > s. \quad (2)$$

Assuming the necessary conditions to assure that at every round the number of queued packets is positive, independently of the relation between $|\alpha|$ and $|\beta|$, and also independently of which are the values of those lengths, the statement can hold for a big enough s , i.e., an injection rate r can be found such that the inequality in (2) holds. ■

Lemma 7. *The digraphs in $\mathcal{E}(\mathcal{U}_1^1)$ are not stable under FIFO.*

Proof. We first prove that the system $(\mathcal{U}_1^1, \text{FIFO})$ can be made unstable. Then, we show that the systems with a network which is a proper extension of \mathcal{U}_1^1 , can also be made unstable under the FIFO protocol.

Digraph \mathcal{U}_1^1 : At the beginning there are s packets queued in e_3 requiring to traverse only edge e_3 . Then the adversary will play infinitely the following rounds:

Round 1: for s steps, the adversary injects rs packets of the form (e_3fe_1) , which get all blocked.

Round 2: for rs steps, the adversary injects r^2s packets of the form (e_3fe_2) and r^2s packets of the form (e_1) . The former are all blocked in e_3 . The latter get mixed with the packets from the previous round and, at the end, there are r^2s packets all with the form (e_1) .

Round 3: for r^2s steps, the adversary injects r^3s packets of the form (e_1) and r^3s packets of the form (fe_2) . The former get all blocked in e_1 while the latter get mixed with the packets flowing from e_3 . From that mixture, a total amount of r^3s packets of the form (fe_2) are remaining.

Round 4: for r^3s steps, the adversary injects r^4s packets of the form (e_1fe_3) and r^4s packets of the form (e_2) . The former get all blocked in e_1 while the latter get mixed with packets from the previous round. A total quantity of r^4s requiring only e_2 remain at the end of the round.

Round 5: for r^4s steps, the adversary injects r^5s packets of the form (e_2) . These injections get all blocked. Moreover, the adversary injects r^5s packets of the form (f) and, desynchronized one step in time with these ones, it injects another r^5s packets of the form (e_3) . Since we want to ensure that the fact of desynchronizing one step in time does not hinder to inject the same quantity of packets, i.e., that we do not lose the last one, it is required that $\lfloor r \cdot r^4s \rfloor = \lfloor r \cdot (r^4s + 1) \rfloor$. Then, the single injections in f get mixed with the packets from the previous round. The mixture is composed by $r^6s/(r+1)$ packets of the form (f) and $r^5s/(r+1)$ of the form (fe_3) . The single injections in e_3 get also mixed with the packets from the previous round (because they

are desynchronized with the other single injections), and r^5s packets requiring only e_3 remain queued in that edge.

Round 6: for r^5s steps, the adversary introduces r^6s single injections of the form (e_3) . These injections get mixed with the injections queued from the previous round.

At the end of the sixth round, there are $r^6s + (r^5s/(r+1))$ packets queued in e_3 and requiring only that edge. The adversary described here achieves that the system $(\mathcal{U}_1^1, \text{FIFO})$ is not stable whether $r^5 + r^6 + r^7 > r + 1$. This holds for any $r \geq 0.929$.

Digraphs in $\mathcal{E}(\mathcal{U}_1^1)$: Let \mathcal{G} be a proper extension of \mathcal{U}_1^1 , i.e., a $\mathcal{G} \in \mathcal{E}(\mathcal{U}_1^1) \setminus \mathcal{U}_1^1$ described at Figure 4(b).⁸ Let us denote with a dashed style not an arc but a directed path. Arcs are labelled with latin letters, while paths are labelled with greek letters. In $\mathcal{G} \in \mathcal{E}(\mathcal{U}_1^1) \setminus \mathcal{U}_1^1$, let $|\alpha| \geq 1$, $|\beta| \geq 1$, $|\gamma| \geq 1$, $|\delta| \geq 1$. We define an adversary with an strategy analogous to the adversary used for \mathcal{U}_1^1 by extending the trajectories of the packets according to the corresponding current paths.

At the beginning there are s packets queued in the first edge of δ , which require to traverse the path (δe_3) . Then the adversary will play infinitely the following rounds:

Round 1: for s steps, the adversary injects rs packets of the form $(\delta e_3 \alpha f \beta e_1)$, which get all blocked at the first edge of the path δ .

Round 2: for rs steps, the adversary injects r^2s packets of the form $(\delta e_3 \alpha f \gamma e_2)$ and r^2s packets of the form (βe_1) . The former are all blocked at the first edge of the path δ . The latter get mixed with the packets from the previous round and, at the end, there are $r^2s - r(|\delta| + |\alpha|)$ packets all waiting to traverse the path (βe_1) .

Round 3: for r^2s steps, the adversary injects r^3s packets of the form (βe_1) and r^3s packets of the form $(\alpha f \gamma e_2)$. From the former, $r^3s - r(|\delta| + |\alpha|)$ remain blocked at the begining of the path β , while the latter get mixed with the packets from the previous round flowing trough δ and e_3 . From that mixture, a total amount of $r^3s - r|\delta|$ packets of the form $(\alpha f \gamma e_2)$ will remain at the beginning of the path α .

Round 4: for $r^3s - r|\delta|$ steps, the adversary injects $r^4s - r^2|\delta|$ packets of the form $(\beta e_1 \alpha f \delta e_3)$ and $r^4s - r^2|\delta|$ packets of the form (γe_2) . From the former, $r^4s - r^2|\delta| - r(2|\delta| + |\alpha|)$ packets remain blocked at the beginning of β , while the latter get mixed with packets from the previous round flowing from α through f . A total quantity of $r^4s - r^2|\delta| - r|\alpha|$ requiring to traverse the path (γe_2) remain queued in the first edge of γ at the end of the round.

Round 5: for $r^4s - r^2|\delta| - r|\alpha|$ steps, the adversary injects $r^5s - r^3|\delta| - r^2|\alpha|$ packets of the form (γe_2) . These injections get all blocked at the origin. Moreover, the adversary injects $r^5s - r^3|\delta| - r^2|\alpha|$ packets of the form (αf) and, desynchronized one step in time with these ones, it injects another $r^5s - r^3|\delta| - r^2|\alpha|$ packets of the form (δe_3) .

The injections with path (αf) get mixed with the packets from the previous round. The mixture is composed by $(r(r^5s - r^3|\delta| - r^2|\alpha| - r|\beta|))/(r+1)$ packets of the form (αf) and $(r^4s - r^2|\delta| - r(2|\delta| + |\alpha|) - |\beta|)/(r+1)$ of the form $(\alpha f \delta e_3)$. The injections in δe_3 get also mixed with the packets from the previous round (because they are desynchronized with the other single

⁸ Observe that the digraphs that result from 2-cycle subdivisions of \mathcal{U}_1^1 contain \mathcal{U}_2 as a subgraph, and so they can be made unstable under the FIFO protocol by an adversary that only uses the \mathcal{U}_2 subgraph and uses an strategy as described in Lemma 6.

injections), and $r^4s - r^2|\delta| - r(2|\delta| + |\alpha|) - |\beta| - |\alpha| - ((r^5s - r^3|\delta| - r^2|\alpha| - r^2|\beta|)/(r+1))$ packets⁹ requiring to traverse still the path (δe_3) remain queued at the begining of δ .

Round 6: for r^5s steps, the adversary introduces r^6s single injections of the form (δe_3) . These injections get mixed with the injections queued from the previous round.

At the end of the sixth round, there are $r^4s - r^2|\delta| - r(2|\delta| + |\alpha|) - |\beta| - |\alpha| - ((r^5s - r^3|\delta| - r^2|\alpha| - r^2|\beta|)/(r+1)) + r^6s - r^5s$ packets queued at the begining of δ which require to traverse δ in order to reach their destination at edge e_3 . The adversary described above only uses path trajectories and makes any digraph $\mathcal{G} \in \mathcal{E}(\mathcal{U}_1^1) \setminus \mathcal{U}_1^1$ unstable when

$$r^4s - r^2|\delta| - r(2|\delta| + |\alpha|) - |\beta| - |\alpha| - \frac{r^5s - r^3|\delta| - r^2|\alpha| - r^2|\beta|}{r+1} + r^6s - r^5s > s. \quad (3)$$

Note that the relations between $|\beta|$, $|\gamma|$ and $|\delta|$ are not important for the statement to hold because the role of the paths (δe_3) , (γe_2) and (βe_1) can be exchange indistinctly. Assuming the necessary conditions to assure that at every round the number of queued packets is positive, independently of the relation of $|\alpha|$ with the length of the other paths β , γ and δ , and also independently of which are the values of those lengths, the statement can hold for a big enough s , i.e., an injection rate r can be found such that the inequality in (3) holds. ■

Lemma 8. *The digraphs in $\mathcal{E}(\mathcal{U}_1^2)$ are not stable under FIFO.*

Proof. Since we consider paths in which every vertex can be visited more than once but every edge can only be traversed once, then the same adversaries as the ones described in Lemma 6 for the digraph \mathcal{U}_2 and the family $\mathcal{E}(\mathcal{U}_2)$ can make any system $(\mathcal{G}, \text{FIFO})$, where $\mathcal{G} \in \mathcal{E}(\mathcal{U}_1^2)$, not to be stable. ■

Lemma 9. *The digraphs in $\mathcal{E}(\mathcal{U}_1^3, \text{FIFO})$ are not stable under FIFO.*

Proof. We first prove that the system $(\mathcal{U}_1^3, \text{FIFO})$ can be made unstable. Then, we show that the systems with a network which is a proper extension of \mathcal{U}_1^3 , can also be made unstable under the FIFO protocol.

Digraph \mathcal{U}_1^3 : At the beginning there are s packets queued in f_1 requiring to traverse only edge f_1 . Then the adversary will play infinitely the following rounds:

Round 1: for s steps, the adversary injects rs packets of the form $(f_1 f_2 e_2)$. These injections will all get blocked in f_1 .

Round 2: for rs steps, the adversary injects r^2s packets of the form $(f_1 f_2 e_1)$ and r^2s packets of the form (e_2) . Since the packets injected in the previous round flow now, the former will remain all blocked in f_1 and the latter will mix in e_2 . From this mix, r^2s packets of the form (e_2) will remain in the queue of e_2 .

Round 3: for the next r^2s steps, the adversary injects r^3s packets of the form (e_2) and r^3s packets of the form $(f_2 e_1)$. The former will be blocked in e_2 by the packets there from the

⁹ $((r^5s - r^3|\delta| - r^2|\alpha| - r^2|\beta|)/(r+1))$ is the quantity of injections of the form (αf) that flow through, i.e., that are not blocked at their origin.

previous round. The latter will mix at f_2 with the packets flowing from the previous round and, at the end, r^3s of them will remain queued in f_2 with path (f_2e_1) .

Round 4: for the next r^3s steps, the adversary injects r^4s packets of the form (e_2) and r^4s packets of the form $(f_2e_1f_1)$. Both sets of injections blocked in the first edge of their paths by the queued injections from the previous round.

Round 5: for the next r^4s steps, the adversary injects r^5s packets of the form (e_2f_1) and r^5s packets of the form (e_1) . The former remain queued in e_2 . The latter mix in e_1 with the injections from the previous round (which are flowing in this round), and queue a total amount of r^5s packets in the following proportions: $\frac{r^6}{r+1}s$ of the form (e_1) , and $\frac{r^5}{r+1}s$ of the form (e_1f_1) .

Round 6: for the next r^5s steps, the adversary injects r^6s packets of the form (f_1) . These injections get mixed with the injections queued from the previous round.

At the end of the sixth round, there are $r^6s + \frac{r^5s}{r+1}$ packets queued in f_1 and requiring only that edge. The adversary described here achieves that the system $(\mathcal{U}_1^3, \text{FIFO})$ is not stable whether $(r^6 + \frac{r^5}{r+1})s > s$, i.e., whether $r^5 + r^6 + r^7 > r + 1$. This holds for any $r \geq 0.929$.

Digraphs in $\mathcal{E}(\mathcal{U}_1^3)$: Let \mathcal{G} be a proper extension of \mathcal{U}_1^3 , i.e., a $\mathcal{G} \in \mathcal{E}(\mathcal{U}_1^3) \setminus \mathcal{U}_1^3$ described at Figure 4(b).¹⁰ Let us denote with a dashed style not an arc but a directed path. Arcs are labelled with latin letters, while paths are labelled with greek letters. In $\mathcal{G} \in \mathcal{E}(\mathcal{U}_1^3) \setminus \mathcal{U}_1^3$, let $|\alpha| \geq 1$, $|\beta| \geq 1$, and $|\gamma| \geq 1$. We define an adversary with an strategy analogous to the adversary used for \mathcal{U}_1^3 by extending the trajectories of the packets according to the corresponding current paths.

At the beginning there are s packets queued in f_1 requiring to traverse only edge f_1 . Then the adversary will play infinitely the following rounds:

Round 1: for s steps, the adversary injects rs packets of the form $(f_1\alpha f_2\gamma e_2)$. These injections will all get blocked in f_1 .

Round 2: for rs steps, the adversary injects r^2s packets of the form $(f_1\alpha f_2\beta e_1)$ and r^2s packets of the form (γe_2) . Since the packets injected in the previous round flow now, the former will remain all blocked in f_1 and the latter will mix in the first edge of γ . From this mix, $r^2s - r|\alpha|$ packets with trajectory (γe_2) will remain in the queue of the first edge in γ .

Round 3: for the next r^2s steps, the adversary injects r^3s packets of the form (γe_2) and r^3s packets of the form $(f_2\beta e_1)$. From the former, $r^3s - r|\alpha|$ will be blocked in the first edge of γ by the packets there from the previous round. The latter will mix at f_2 with the packets flowing from the previous round and, at the end, $r^3s - r|\alpha|$ of them will remain queued in f_2 with trajectory $(f_2\beta e_1)$.

Round 4: for the next $r^3s - r|\alpha|$ steps, the adversary injects $r^4s - r^2|\alpha|$ packets of the form (γe_2) and $r^4s - r^2|\alpha|$ packets of the form $(f_2\beta e_1f_1)$. Both sets of injections blocked in the first edge of their paths by the queued injections from the previous round.

Round 5: for the next $r^4s - r^2|\alpha|$ steps, the adversary injects $r^5s - r^3|\alpha|$ packets of the form (γe_2f_1) and $r^5s - r^3|\alpha|$ packets of the form (βe_1) . The former remain queued in the first edge of γ . The latter mix in the first edge of β with the injections from the previous round (which

¹⁰ Observe that the digraphs that result from 2-cycle subdivisions of \mathcal{U}_1^3 contain \mathcal{U}_2 as a subgraph, and so they can be made unstable under the FIFO protocol by an adversary that only uses the \mathcal{U}_2 subgraph and uses an strategy as described in Lemma 6.

are flowing in this round), and queue a total amount of $r^5s - r^3|\alpha|$ packets in the following proportions: $(r^6s - r^4|\alpha|)/(r+1)$ of the form (βe_1) , and $(r^5s - r^3|\alpha|)/(r+1)$ of the form $(\beta e_1 f_1)$.

Round 6: for the next $r^5s - r^3|\alpha|$ steps, the adversary injects $r^6s - r^4|\alpha|$ packets of the form (f_1) . These injections get mixed with the injections queued from the previous round.

At the end of the sixth round, there are more than $((r^5s - r^3|\alpha|)/(r+1)) + r^6s - r^4|\alpha| - r(|\beta| + |\gamma|)$ packets queued in f_1 , which require to cross that edge. The adversary described above only uses path trajectories and makes any digraph $\mathcal{G} \in \mathcal{E}(\mathcal{U}_1^3) \setminus \mathcal{U}_1^3$ unstable when

$$\frac{r^5s - r^3|\alpha|}{r+1} + r^6s - r^4|\alpha| - r(|\beta| + |\gamma|) > s. \quad (4)$$

Note that the relation between $|\beta|$ and $|\gamma|$ is not important for the statement to hold because the role of the paths (βe_1) and (γe_2) can be exchange indistinctly. Assuming the necessary conditions to assure that at every round the number of queued packets is positive, independently of the relation of $|\alpha|$ with the length of the other paths β and γ , and also independently of which are the values of those lengths, the statement can hold for a big enough s , i.e., an injection rate r can be found such that the inequality in (4) holds. ■

Lemma 10. *The digraphs in $\mathcal{E}(\mathcal{U}_1^4)$ are not stable under FIFO.*

Proof. We first prove that the system $(\mathcal{U}_1^4, \text{FIFO})$ can be made unstable. Then, we show that the systems with a network which is a proper extension of \mathcal{U}_1^4 , can also be made unstable under the FIFO protocol.

Digraph \mathcal{U}_1^4 : The digraph \mathcal{U}_1^4 was already shown in [11] not to be stable under FIFO, although in there was a mistake in the counting of the last two rounds of the inductive reasoning (and thus, on the resulting injection rate). Doing the right calculations, and with the same adversary as in [11], we get that the digraph \mathcal{U}_1^4 can be made unstable when $(r^7s + r^6s + r^5s)/(r+1) > s$, i.e., for injection rate $r \geq 0.929$.

Digraphs in $\mathcal{E}(\mathcal{U}_1^4)$: Let \mathcal{G} be a proper extension of \mathcal{U}_1^4 , i.e., a $\mathcal{G} \in \mathcal{E}(\mathcal{U}_1^4) \setminus \mathcal{U}_1^4$ described at Figure 4(b).¹¹ Let us denote with a dashed style not an arc but a directed path. Arcs are labelled with latin letters, while paths are labelled with greek letters. In $\mathcal{G} \in \mathcal{E}(\mathcal{U}_1^4) \setminus \mathcal{U}_1^4$, let $|\alpha| \geq 1$, $|\beta| \geq 1$, and $|\gamma| \geq 1$. We define an adversary with an strategy analogous to the adversary used for \mathcal{U}_1^4 by extending the trajectories of the packets according to the corresponding current paths.

At the beginning there are s packets queued in f requiring to traverse only edge f . Then the adversary will play infinitely the following rounds:

Round 1: for s steps, the adversary injects rs packets of the form $(f\alpha\beta e_1)$. These injections get all blocked at the queue of f .

Round 2: for rs steps, the adversary injects r^2s packets with trajectory $(fae_{21}\gamma e_{22})$, and r^2s injections more with trajectory (βe_1) . The former get all blocked at the queue in f . The latter get mixed with the packets flowing from the previous round; at the end of the round, this mixture

¹¹ Observe that the digraphs that result from 2-cycle subdivisions of \mathcal{U}_1^4 contain \mathcal{U}_2 as a subgraph, and so they can be made unstable under the FIFO protocol by an adversary that only uses the \mathcal{U}_2 subgraph and uses an strategy as described in Lemma 6.

is composed by $r^2s - r|\alpha|$ packets, queued at the first edge of the path β , and with a trajectory (βe_1) .

Round 3: for $r^2s + r^3s - r|\alpha|$ steps, the adversary injects $r^3s + r^4s - r^2|\alpha|$ packets with trajectory (βe_1) , and $r^3s + r^4s - r^2|\alpha|$ packets more with trajectory $(e_{21}\gamma e_{22}f)$. From the former, $r^4s - r^2|\alpha|$ remain blocked at the first edge of β . From the latter, a total amount of $r^4s - r^2|\alpha|$ packets with trajectory $(e_{21}\gamma e_{22}f)$ are queued at e_{21} .

Round 4: for $r^4s - r^2|\alpha|$ steps, the adversary inject $r^5s - r^3|\alpha|$ packets with trajectory (βe_1f) , and another $r^5s - r^3|\alpha|$ packets with trajectory (e_{22}) . The former get all blocked at the first edge of β . The latter get mixed with the packets from the previous round that flow through e_{22} after traversing γ ; a total amount of $r^5s - r^3|\alpha| - r|\gamma|$ packets remain at the end of the round in the queue of e_{22} (from them, $(r^5s - r^3|\alpha| - r|\gamma|)/(r+1)$ packets require to traverse both e_{22} and f , while the rest only requires e_{22}).

Round 5: for $r^5s - r^3|\alpha|$ steps, the adversary injects $r^6s - r^4|\alpha|$ injections that only require to traverse (f) .

At the end of the fifth round, there are $((r^5s - r^3|\alpha| - r|\gamma|)/(r+1)) - |\beta| + r^6s - r^4$ packets queued in f , which require to cross (only) that edge. The adversary described above only uses path trajectories and makes any digraph $\mathcal{G} \in \mathcal{E}(\mathcal{U}_1^4) \setminus \mathcal{U}_1^4$ unstable when

$$\frac{r^5s - r^3|\alpha| - r|\gamma|}{r+1} - |\beta| + r^6s - r^4 > s. \quad (5)$$

Assuming the necessary conditions to assure that at every round the number of queued packets is positive, independently of the relation of $|\alpha|$ with the length of the other paths β and γ , and also independently of which are the values of those lengths, the statement can hold for a big enough s , i.e., an injection rate r can be found such that the inequality in (5) holds. ■

A.2 Simple-path instability under FIFO

For clarity reasons, we split the proof of Lemma 5 in the following two lemmas (Lemmas 11 and 12). Moreover, in each of the lemmas, we first prove that the smallest digraph \mathcal{G} of the family can be made unstable under the FIFO protocol and then, we show that the systems with a network which is a proper extension of it, i.e., $\mathcal{G}' \in \mathcal{E}(\mathcal{G}) \setminus \mathcal{G}$, can also be made unstable under the FIFO protocol. By showing first the instability in the smallest forbidden digraphs, one gets a better intuition on how is the strategy of the adversary and which contention power it has. The strategy used when the network is an extension of the smallest forbidden digraph is usually the natural extension to paths of the strategy used there. The only requirement that the fact of dealing with an network extension imposes is that, the initial configuration of the system has to have enough initial packets (the quantity will depend on the length of the edge extensions) to allow the accumulation.

Lemma 11. *The digraphs in $\mathcal{E}(\mathcal{S}_3)$ are not simple-path stable under FIFO.*

Proof. We first prove that the system $(\mathcal{S}_3, \text{FIFO})$ can be made simple-path unstable. Then, we show that the systems with a network which is a proper extension of \mathcal{S}_3 , can also be made simple-path unstable under the FIFO protocol.

Digraph \mathcal{S}_3 : At the beginning there are s packets queued in e_{21} requiring to traverse only edge e_{21} . Then the adversary will play infinitely the following rounds:

Round 1: for s steps, the adversary injects rs packets of the form $(e_{21}e_{22})$. These injections will all get blocked in e_{21} .

Round 2: for rs steps, the adversary injects r^2s packets of the form $(e_{22}f_1)$. These packets will mix in e_{22} with packets from the previous round. From this mix, $r^3s/(r+1)$ are from the injections of these round and $r^2/(r+1)$ are still packets from the previous round. The important think is tht that the total queue at e_{22} has size r^2s .

Round 3: for the next r^2s steps, the adversary injects r^3s packets of the form $(e_{22}f_1e_1)$. They will be blocked in e_{22} by the packets there from the previous round.

Round 4: for the next r^3s steps, the adversary injects r^4s packets of the form $(e_{22}f_1)$ and r^4s packets of the form (e_1f_{22}) . The former are blocked at e_{22} , while the latter get mixed with the packets flowing from the previous round.

Round 5: for the next r^4s steps, the adversary injects the same type of packets as in the previous round. Thus, at the end there are r^5s packets of the form $(e_{22}f_1)$ and r^5s packets of the form (e_1f_22) .

Round 6: for the next r^5s steps, the adversary injects r^6s packets of the form $(e_{22}f_1)$ and r^6s packets of the form $(f_{22}f_{21}e_{21})$. The former get all blocked. The latter get mixed with the ones flowing from the previous round but the total quantity of packets blocked is r^6s .

Round 7: for the next r^6s steps, the adversary injects r^7s packets of the form (f_1) and r^7s more of the form $(f_{22}f_{21}e_{21})$. The latter get all blocked, while the former get mixed at f_1 with the packets $(e_{22}f_1)$ flowing from the previous round. From this mix, a total amount of r^7s packets will be queued at f_1 .

Round 8: for the next r^7s steps, the adversary injects r^8s packets with path (f_1e_{21}) and r^8s packets with path (f_{21}) . The former get all blocked in f_1 , while the latter get mixed with the ones flowing from the previous round. In this mix there are $r^8s/(r+1)$ packets with path $(f_{22}f_{21}e_{21})$ and $r^9s/(r+1)$ with path (f_{21}) .

Round 9: for the next r^8s steps, the adversary injects r^9s packets with path (e_{21}) .

At the end of the ninth round, there are $\frac{r^8s}{r+1} + r^9s$ packets queued in e_{21} and requiring only that edge. The adversary described here achieves that the system $(\mathcal{S}_3, \text{FIFO})$ is not simple-path stable whether $(r^8 + r^9 + r^{10})s/(r+1) > s$, i.e., whether $r^8 + r^9 + r^{10} > r+1$. This holds for any $r \geq 0.954$.

Digraphs in $\mathcal{E}(\mathcal{S}_3)$: Let \mathcal{G} be a proper extension of \mathcal{S}_3 , i.e., a $\mathcal{G} \in \mathcal{E}(\mathcal{S}_3) \setminus \mathcal{S}_3$ described at Figure 5(b). Let us denote with a dashed style not an arc but a directed path. Arcs are labelled with latin letters, while paths are labelled with greek letters. Moreover $|\alpha| \geq 1$ and $|\gamma| \geq 1$. We define an adversary with an strategy analogous to the adversary used for \mathcal{S}_3 by extending the trajectories of the packets according to the corresponding current paths.

At the beginning there are s packets queued in e_{21} requiring to traverse only edge e_{21} . Then the adversary will play infinitely the following rounds:

Round 1: for s steps, the adversary injects rs packets of the form $(e_{21}e_{22}\alpha)$. These injections will all get blocked in e_{21} .

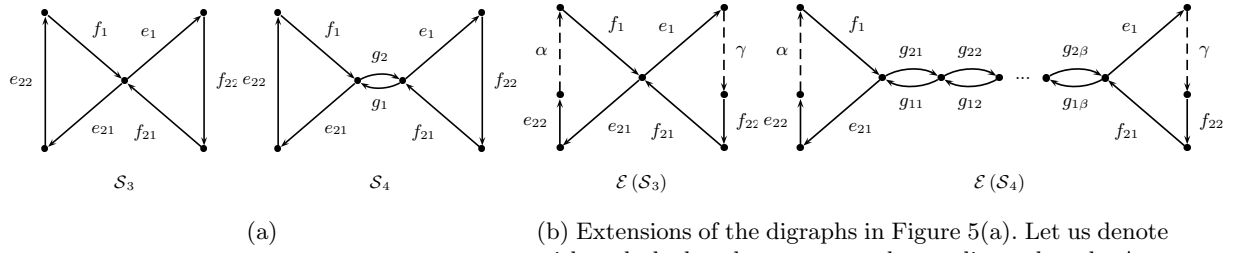


Fig. 5. Members of the family of forbidden subgraphs characterizing simple-path stability under FIFO.

Round 2: for rs steps, the adversary injects r^2s packets of the form $(e_{22}\alpha f_1)$. These packets will mix in e_{22} with packets from the previous round. From this mix, the queue at e_{22} will keep a total amount of r^2s packets.

Round 3: for the next r^2s steps, the adversary injects r^3s packets of the form $(e_{22}\alpha f_1 e_1)$. They will be blocked in e_{22} by the packets there from the previous round.

Round 4: for the next r^3s steps, the adversary injects r^4s packets of the form $(e_{22}\alpha f_1)$ and r^4s packets of the form $(e_1\gamma f_{22})$. The former are blocked at e_{22} , while the latter get mixed with the packets flowing from the previous round, and a total amount of $r^4s - r|\alpha|$ packets are kept in the queue of e_1 .

Round 5: for the next r^4s steps, the adversary injects the same type of packets as in the previous round. Thus, at the end there are r^5s packets of the form $(e_{22}\alpha f_1)$ queued at e_{22} and $r^5s - r|\alpha|$ packets of the form $(e_1\gamma f_{22})$ queued at e_1 .

Round 6: for the next r^5s steps, the adversary injects r^6s packets of the form $(e_{22}\alpha f_1)$ and r^6s packets of the form $(\gamma f_{22} f_{21} e_{21})$. The former get all blocked. The latter get mixed with the ones flowing from the previous round but the total quantity of packets blocked at the first edge of γ is $r^6s - r|\alpha|$.

Round 7: for the next r^6s steps, the adversary injects r^7s packets of the form (f_1) and r^7s more of the form $(\gamma f_{22} f_{21} e_{21})$. From the latter, $r^7s - r|\alpha|$ get blocked, while the former get mixed at f_1 with the packets flowing from the previous round. From this mix, a total amount of $r^7s - r|\alpha|$ packets will remain queued at f_1 .

Round 8: for the next $r^7s - r|\alpha|$ steps, the adversary injects $r^8s - r^2|\alpha|$ packets with path $(f_1 e_{21})$ and $r^8s - r^2|\alpha|$ packets with path (f_{21}) . The former get all blocked in f_1 , while the latter get mixed at f_{21} with the ones flowing from the previous round. In this mix there are $((r^8s - r^2|\alpha|)/(r+1)) - |\gamma|$ packets with path $(f_{21} e_{21})$.

Round 9: for the next $r^8s - r^2|\alpha|$ steps, the adversary injects $r^9s - r^3|\alpha|$ packets with path (e_{21}) .

At the end of the ninth round, there are more than $r^9s - r^3|\alpha| - |\gamma| + ((r^8s - r^2|\alpha|)/(r+1))$ packets queued in e_{21} and requiring to traverse only that edge. The adversary described above

only uses simple-path trajectories and makes any digraph $\mathcal{G} \in \mathcal{E}(\mathcal{S}_3) \setminus \mathcal{S}_3$ unstable when

$$r^9 s - r^3 |\alpha| - |\gamma| + \frac{r^8 s - r^2 |\alpha|}{r + 1} > s. \quad (6)$$

Assuming the necessary conditions to assure that at every round the number of queued packets is positive, independently of the relation between $|\alpha|$ and $|\gamma|$, and also independently of which are the values of those lengths, the statement can hold for a big enough s , i.e., an injection rate r can be found such that the inequality in (6) holds. ■

Lemma 12. *The digraphs in $\mathcal{E}(\mathcal{S}_4)$ are not simple-path stable under FIFO.*

Proof. We first prove that the system $(\mathcal{S}_4, \text{FIFO})$ can be made simple-path unstable. Then, we show that the systems with a network which is a proper extension of \mathcal{S}_4 , can also be made simple-path unstable under the FIFO protocol. The strategies used by the adversaries to achieve instability are similar to those used in Lemma 11 to show the instability of any digraphs in $\mathcal{E}(\mathcal{S}_3)$ under FIFO.

Digraph \mathcal{S}_4 : At the beginning there are s packets queued in e_{21} requiring to traverse only edge e_{21} . Then the adversary will play infinitely the following rounds:

Round 1: for s steps, the adversary injects rs packets of the form $(e_{21}e_{22})$. These injections will all get blocked in e_{21} .

Round 2: for rs steps, the adversary injects r^2s packets of the form $(e_{22}f_1)$. These packets will mix in e_{22} with packets from the previous round. From this mix, $r^3s/(r+1)$ are from the injections of these round (with path $(e_{22}f_1)$) and $r^2/(r+1)$ are still packets from the previous round (with path (e_{22})). The total queue at e_{22} has size r^2s .

Round 3: for the next r^2s steps, the adversary injects r^3s packets of the form $(e_{22}f_1g_2e_1)$. They will be blocked in e_{22} by the packets there from the previous round.

Round 4: for the next r^3s steps, the adversary injects r^4s packets of the form $(e_{22}f_1g_2)$ and r^4s packets of the form (e_1f_{22}) . The former are blocked at e_{22} , while the latter get mixed with the packets flowing from the previous round. The queue at e_1 has total size r^4s .

Round 5: for the next r^4s steps, the adversary injects the same type of packets as in the previous round. Thus, at the end there are r^5s packets of the form $(e_{22}f_1g_2)$ and r^5s packets of the form (e_1f_{22}) .

Round 6: for the next r^5s steps, the adversary injects r^6s packets of the form $(e_{22}f_1g_2)$ and r^6s packets of the form $(f_{22}f_{21}g_1e_{21})$. The former get all blocked. The latter get mixed with the ones flowing from the previous round but the total quantity of packets blocked at f_{22} is r^6s .

Round 7: for the next r^6s steps, the adversary injects r^7s packets of the form (f_1) and r^7s more of the form $(f_{22}f_{21}g_1e_{21})$. The latter get all blocked, while the former get mixed at f_1 with the packets $(e_{22}f_1g_2)$ flowing from the previous round. From this mix, a total amount of r^7s packets will be queued at f_1 .

Round 8: for the next r^7s steps, the adversary injects r^8s packets with path (f_1e_{21}) and r^8s packets with path (f_{21}) . The former get all blocked in f_1 , while the latter get mixed with the

ones flowing from the previous round. In this mix there are $r^8 s / (r + 1)$ packets with path $(f_{22} f_{21} g_1 e_{21})$ and $r^9 s / (r + 1)$ with path (f_{21}) .

Round 9: for the next $r^8 s$ steps, the adversary injects $r^9 s$ packets with path (e_{21}) .

At the end of the ninth round, there are $\frac{r^8 s}{r+1} + r^9 s$ packets queued in e_{21} and requiring only that edge. The adversary described here achieves that the system $(\mathcal{S}_4, \text{FIFO})$ is not simple-path stable whether $(r^8 + r^9 + r^{10})s / (r + 1) > s$, i.e., whether $r^8 + r^9 + r^{10} > r + 1$. This holds for any $r \geq 0.954$.

Digraphs in $\mathcal{E}(\mathcal{S}_4)$: Let \mathcal{G} be a proper extension of \mathcal{S}_4 , i.e., a $\mathcal{G} \in \mathcal{E}(\mathcal{S}_4) \setminus \mathcal{S}_4$ described at Figure 5(b). Let us denote with a dashed style not an arc but a directed path. Arcs are labelled with latin letters, while paths are labelled with greek letters. Moreover $|\alpha| \geq 1, |\beta| \geq 1$, and $|\gamma| \geq 1$. In the middle there are $1 + |\beta|$ 2-cycles, each one sharing one vertex with the (unique) previous and the other vertex with the (unique) next. We define an adversary with an strategy analogous to the adversary used for \mathcal{S}_4 by extending the trajectories of the packets according to the corresponding current paths.

At the beginning there are s packets queued in e_{21} requiring to traverse only edge e_{21} . Then the adversary will play infinitely the following rounds:

Round 1: for s steps, the adversary injects rs packets of the form $(e_{21} e_{22} \alpha)$. These injections will all get blocked in e_{21} .

Round 2: for rs steps, the adversary injects $r^2 s$ packets of the form $(e_{22} \alpha f_1)$. These packets will mix in e_{22} with packets from the previous round. From this mix, the queue at e_{22} will keep a total amount of $r^2 s$ packets.

Round 3: for the next $r^2 s$ steps, the adversary injects $r^3 s$ packets with path trajectory $(e_{22} \alpha f_1 g_{21} \dots g_{2\beta} e_1)$. They will be blocked in e_{22} by the packets there from the previous round.

Round 4: for the next $r^3 s$ steps, the adversary injects $r^4 s$ packets of the form $(e_{22} \alpha f_1 g_{21} \dots g_{2\beta})$ and $r^4 s$ packets of the form $(e_1 \gamma f_{22})$. The former are blocked at e_{22} , while the latter get mixed with the packets flowing from the previous round, and a total amount of $r^4 s - r(|\alpha| + |\beta|)$ packets are kept in the queue of e_1 .

Round 5: for the next $r^4 s$ steps, the adversary injects the same type of packets as in the previous round. Thus, at the end there are $r^5 s$ packets of the form $(e_{22} \alpha f_1 g_{21} \dots g_{2\beta})$ queued at e_{22} and $r^5 s - r(|\alpha| + |\beta|)$ packets of the form $(e_1 \gamma f_{22})$ queued at e_1 .

Round 6: for the next $r^5 s$ steps, the adversary injects $r^6 s$ packets of the form $(e_{22} \alpha f_1 g_{21} \dots g_{2\beta})$ and $r^6 s$ packets of the form $(\gamma f_{22} f_{21} g_{1\beta} \dots g_{11} e_{21})$. The former get all blocked. The latter get mixed with the ones flowing from the previous round but the total quantity of packets blocked at the first edge of γ is $r^6 s - r(|\alpha| + |\beta|)$.

Round 7: for the next $r^6 s$ steps, the adversary injects $r^7 s$ packets of the form (f_1) and $r^7 s$ more of the form $(\gamma f_{22} f_{21} g_{1\beta} \dots g_{11} e_{21})$. From the latter, $r^7 s - r(|\alpha| + |\beta|)$ get blocked, while the former get mixed at f_1 with the packets flowing from the previous round. From this mix, a total amount of $r^7 s - r|\alpha|$ packets will remain queued at f_1 .

Round 8: for the next $r^7 s - r|\alpha|$ steps, the adversary injects $r^8 s - r^2 |\alpha|$ packets with path $(f_1 e_{21})$ and $r^8 s - r^2 |\alpha|$ packets with path (g_{11}) . The former get all blocked in f_1 , while the latter get mixed at g_{11} with the ones flowing from the previous round. In this mix there are $((r^8 s - r^2(|\alpha| + |\beta|)) / (r + 1)) - (|\beta| + |\gamma|)$ packets with path $(g_{11} e_{21})$.

Round 9: for the next $r^8 s - r^2 |\alpha|$ steps, the adversary injects $r^9 s - r^3 |\alpha|$ packets with path (e_{21}) .

At the end of the ninth round, there are more than $r^9 s - r^3 |\alpha| - |\beta| - |\gamma| + (r^8 s - r^2 (|\alpha| + |\beta|)) / (r + 1)$ packets queued in e_{21} and requiring to traverse only that edge. The adversary described above only uses simple-path trajectories and makes any digraph $\mathcal{G} \in \mathcal{E}(\mathcal{S}_4) \setminus \mathcal{S}_4$ unstable when

$$r^9 s - r^3 |\alpha| - |\beta| - |\gamma| + \frac{r^8 s - r^2 (|\alpha| + |\beta|)}{r + 1} > s. \quad (7)$$

Assuming the necessary conditions to assure that at every round the number of queued packets is positive, independently of the relation between $|\alpha|$, $|\beta|$ and $|\gamma|$, and also independently of which are the values of those lengths, the statement can hold for a big enough s , i.e., an injection rate r can be found such that the inequality in (7) holds. ■